Planning a Project Using the Oracle Unified Method (OUM) – *An Iterative and Incremental Approach*

An Oracle White Paper
February 2011

Planning a Project Using the Oracle Unified Method (OUM)

Planning a Project Using the Oracle Unified Method (OUM)

"Life belongs to the living, and he who lives must be prepared for changes."

- ***Johann Wolfgang von Goethe***

EXECUTIVE OVERVIEW

Oracle Unified Method (OUM) uses the Manage Focus Area - Project Management Method (OUM Manage) to provide a framework in which projects can be planned, estimated, controlled, and tracked in a consistent manner. OUM is based on the Unified Process and is therefore both iterative and incremental. The intent of this white paper is to provide you with a high-level guide for planning an OUM project.

INTRODUCTION

Iterative development using OUM provides many benefits: major risks are identified and addressed early in the project; requirement changes are identified and prioritized efficiently; project team utilization is optimized; and progress and quality are continuously monitored and corrected.

PLANNING AN OUM PROJECT – A LAYERED APPROACH

OUM is designed to be a scalable method that can be applied to any kind of Oracle project. Plans need to be scalable for different project sizes and complexity and contain the right level of detail for the current planning horizon. Plans that are too detailed are almost instantly inaccurate and obscure key objectives. On the other hand, plans that are too high level will not allow for measurement of project progress nor keep the project team focused on their day-to-day activities.

Plans need to display the appropriate level of detail and planning horizon for specific audiences. For example, C-level executives, business area managers and external stakeholders are rarely interested in the fine details of the project. Their focus is on the release date, major milestones, business impacts, points at which major decisions must be made. On the other hand, the project team needs the details on the lower level plans to plan their daily work and measure progress.

Therefore, a layered approach to planning OUM projects is recommended. As shown in Figure 1 below, there are two plans active in the project at any given time on an OUM project – the implementation and the iteration plan.

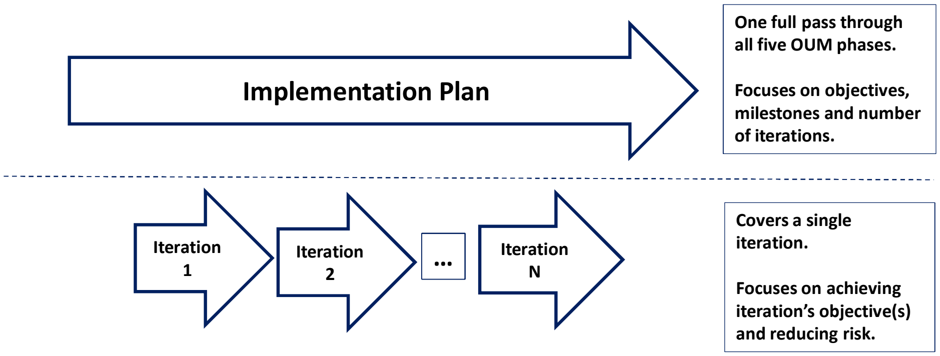

Implementation Plan
One full pass through all five OUM phases.
Focuses on objectives, milestones and number of iterations.
Iteration 1
Iteration 2
...
Iteration N
Covers a single iteration.
Focuses on achieving iteration's objective(s) and reducing risk.

Figure 1: OUM Planning Layers

DEFINITION OF PLANS AT EACH LAYER

The Implementation Plan

The implementation plan is a brief, coarse-grained outline for the project. The objectives for the project are reflected in the implementation plan and tie back to the Business Case developed in the OUM Envision Focus Area. The main purpose of the implementation plan is to provide a roadmap for achieving the project's goals, along with a sketch of the number and purpose of each iteration needed per phase.

The Iteration Plan

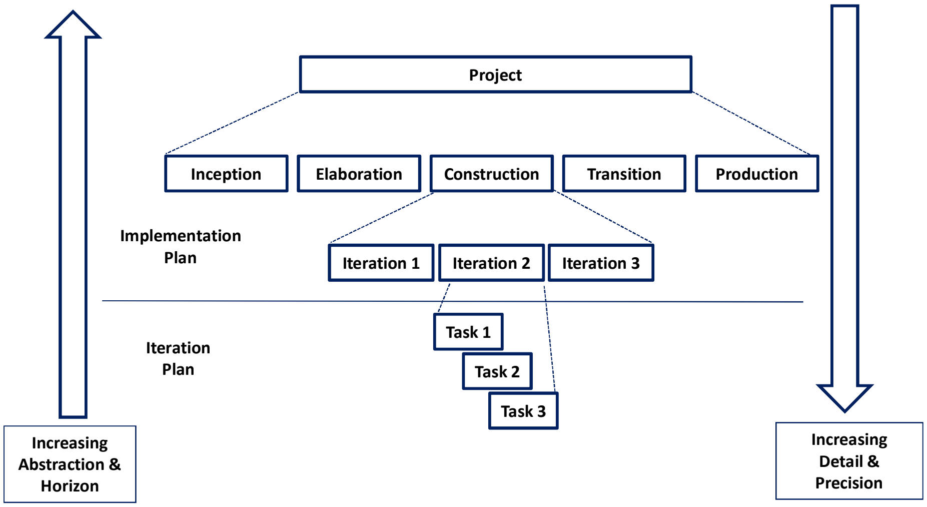
The iteration plan represents the lowest and most detailed layer of planning. An individual iteration plan will be created for each iteration in the project. The current iteration plan focuses on how an incremental set of objectives and/or functionality will be delivered within the framework of the project. The main purpose of the iteration plan is to lay out how the team will achieve the stated objectives for the given iteration.

"Agile planning is focused more on the planning than on the creation of the plan, encourages change, results in plans that are easily changed, and is spread throughout the project."

- ***Mike Cohn, "Agile estimating and planning" (Pearson Education, Inc.)***

PLANNING USING A TOP-DOWN/BOTTOM-UP APPROACH

In OUM, plans are created using a top-down/bottom-up approach in which the plans at each layer (implementation and iteration) are created and maintained in a simultaneous fashion. Because the plans are inter-related, it is likely that a change to a plan at one layer will cause the need for an adjustment to a plan at another layer. The planning layers are show in Figure 2 below.



Adapted from Kurt Bittner and Ian Spence, "Managing iterative software development projects" (Pearson Education, Inc.)

Figure 2: Top-Down/Bottom Up Planning

The planning starts in the OUM Manage Project Start Up phase where the focus is on the implementation plan. Also in Project Start Up, the initial iteration of the OUM Implement Inception phase is drafted to the degree that the project is able to move into the Inception phase.

During Inception, the implementation plan created in Project Start Up is further refined as more project requirements and risks are uncovered. As the project iterates through the later phases, the plans at each layer (implementation and iteration) will be adjusted based on the results of iteration assessments and as project objectives shift, as is often the case.

On any OUM project, an iteration plan for the upcoming iteration is created before that iteration begins. Also, the implementation plan must be examined to ensure it is still valid as the iteration plans are created and maintained.

Therefore, at any time on an OUM project the following plans should be active:

- One implementation plan
- Two iteration plans – one for the current iteration and a draft for the upcoming iteration

IMPLEMENTATION PLANNING

In OUM, the implementation plan is an outline of the project, showing the total number of planned iterations across the five OUM phases (Inception, Elaboration, Construction, Transition and Production) as well as key milestone dates for each of these iterations.   The implementation plan will sketch out the work across the five phases by outlining the number of iterations in each phase and major milestones, as shown in the example in Figure 3 below.

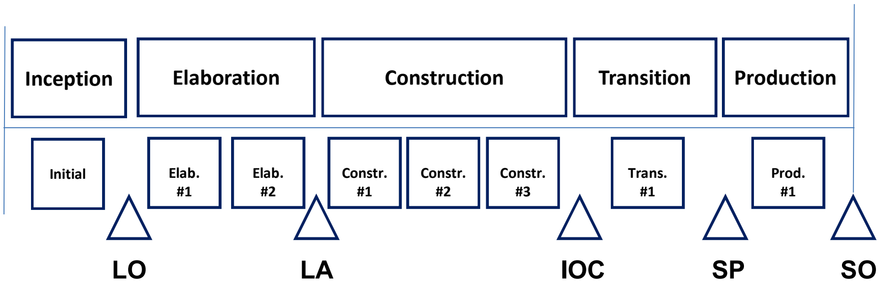


Figure 3: Example Implementation Plan

Keep in mind, that because OUM utilizes a top-down/bottom-up approach to planning, the task of developing the implementation plan requires the implementation plan and current iteration plan to be worked on concurrently.

Compile the Implementation Plan

The implementation plan presents a roadmap of the planned iterations and should summarize the following:

- Objective(s) of each iteration and its contribution to the project goal(s)
- Business milestones
- Project commitments
- Team composition
- Project dependencies
- Project approach

Factors that Drive the Number of Iterations per Phase

As part of the implementation planning, the number of iterations per phase will need to be determined. In general, OUM recommends using the standard number of iterations as depicted in the OUM Implement Focus Area diagram. However, there are several factors that can increase or decrease the number of iterations represented in the plan(s). The factors which would increase the number of iterations per phase are shown in Table 1 below.

Phase	Factors
Inception	• Highly variable scope • Lack of artifacts from Envision or Envision not conducted • Poorly defined/documented business objectives • Disagreement between stakeholders
Elaboration	• Unstable architecture • Unproven architecture • Unstable requirements • Unavailable development environment • Challenging non-functional requirements
Construction	• Large amounts of functionality to develop and test • Poor architecture • Limited team resources
Transition	• Hardware replacement or rollout • Number of deployment and/or rollout sites • Large numbers of users to be trained or complex

	training required
Production	• Level and length of support required • Incremental rollout strategy

Table 1: Iteration Factors

Secure Agreement on the Implementation Plan

Before commencing work, present the plan to the stakeholders to gain agreement on the implementation plan. The project team in particular should agree on the implementation plan.

Evolve the Implementation Plan

Keep in mind that implementation planning, as with the other planning efforts, is done in an iterative manner. Information will be uncovered during the planning process that will necessitate amending the implementation plan as well as the current iteration plan. As the project progresses phase and iteration assessments will be conducted and feedback will be gained, which will lead to the need to adjust the plan(s).

ITERATION PLANNING

Iteration planning is where the bulk of the planning for a project occurs. Each iteration should be planned such that a set of specific objectives are accomplished and a group of project risks are addressed. A project manager will typically analyze and manage the current iteration plan on a daily basis.

> **"This progressive detailing of the project management plan is often called "rolling wave planning", indicating that planning and documentation is an iterative and ongoing process."**
>
> - ***A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fourth Edition***

Assess the Current State of Project Risks

Risk management on any project is a dynamic process since project risks will be eliminated and additional risks will be exposed as the project progresses.

For each iteration, the top risks should be identified based on the current state of the project (i.e., progress to-date, project phase, team experience, etc.). A rule of thumb is to narrow the top risks to a list of no more than 10. Based on the list of the top 10 risks, the iteration plan can be developed to actively address each risk.

Establish the Scope of the Current Iteration

The scope of the iteration will be determined according to the objectives for the current iteration, team capacity and the iteration's deliverables. These factors are discussed in the sections below.

Objectives of the Current Iteration

Based upon the results of the risk analysis and the progress of the project up to this point must be refined to provide a clear set of measurable and achievable objectives for the given iteration. The objectives for any iteration should be based on the current phase the project and should relate to meeting the milestone objectives for the particular phase.

Team Capacity

The team's capacity is a broad measure of the amount of effort the team will be able to take on in the iteration. The team capacity must be considered when planning the scope of work for the iteration. The capacity is determined by the team's size, availability and velocity, which refers to the speed at which a team can implement and test use cases and change requests.

As the project progresses, it is often the case that requirements are handled with less effort which results in an increase in the team's velocity. Velocity can be used to develop the initial estimates of the duration and effort required to complete the current iteration.

A team's velocity over time can be tracked using a burn-down Chart.

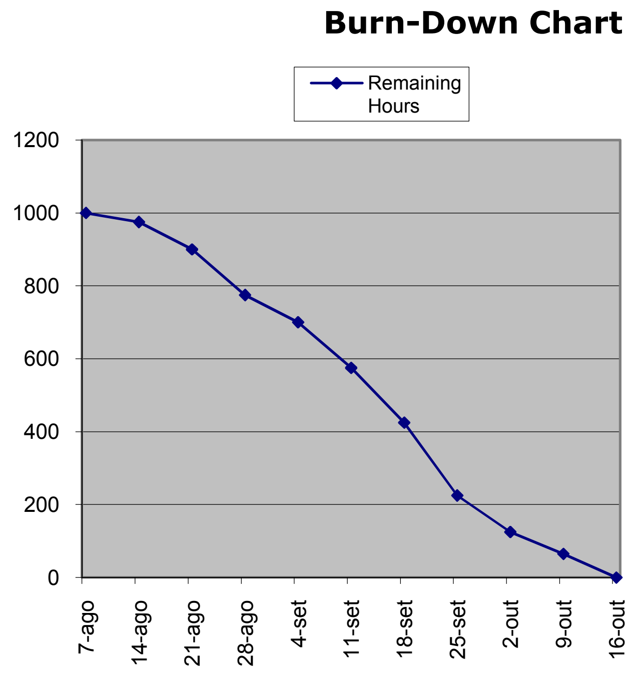


Figure 4: Burn-Down Chart

The burn-down chart in Figure 4 above tracks progress by showing a decrease in the remaining hours. A project team can select any measure they deem relevant to measure progress such as the number of scenarios completed, days remaining, etc.

The current team's capacity drives the initial estimates for the effort and duration for the given iteration. These initial estimates are needed to select which objectives will be included in the iteration plan and will be further refined as the iteration planning progresses.

"You must ensure that the iteration objectives explicitly address the risks currently facing the project: the purpose of an iteration is to mitigate a particular set of risks."

- ***Kurt Bittner and Ian Spence, "Managing iterative software development projects" (Pearson Education, Inc.)***

.

Deliverables for the Iteration

The deliverables should be associated with one or more of the iteration's objectives. The iteration's deliverables should not be confused with the OUM task work products. The work products are just a means to an end to producing the deliverable and are not necessarily related to the project's objectives.

The most important deliverable of any iteration is the increment that will be developed. This is defined in OUM by the set of use cases and other non-functional requirements that will be achieved during the iteration. These requirements are selected from the project's current MoSCoW list. Selecting the right set of requirements requires collaboration among all team members and should be focused on the current M's and S's shown on the MoSCoW list. Simply stated, the MoSCoW List registers the known business requirements. Each requirement is classified by the first letter of: Must, Should, Could or Won't.

As the project moves along and risks are retired, the focus of the iterations can shift to filling out the needed functionality for the product under development. Also, change requests and identified defects can be addressed as the more architecturally significant use cases have been realized. To achieve the right balance the required functionality, change requests and defects must be prioritized and estimated by analyzing the MoSCoW List. See Task RD.045 - Prioritize Requirements for more information on how MoSCoW is used in OUM.

Map the Iteration's Deliverables to OUM Tasks

The iteration plan needs lay out the detailed tasks required to achieve the scope that has been established for the current iteration. This is done by selecting the appropriate tasks from the OUM WBS for the current phase of the project.

As with any OUM project, it is recommended that the project manager build the project workplan by starting with the appropriate pre-tailored view for the type of project under consideration (Solution-Driven Application Implementation, Requirements-Driven Application Implementation, etc.) and scaling the project for the given circumstances. The project workplan should focus on the tasks included in the OUM Implement Core Workflow, which identifies the core tasks within the Implement Focus Area.

Assign Resources and Produce Estimates

Resources need to be associated with each OUM task included in the iteration plan. The resources are assigned based on availability and matching the appropriate skill-set to the task. All OUM tasks contain a Roles and Responsibilities table that can be used when assigning resources to tasks. In addition, the Project Roles reference page provides additional information about the specifics of each role.

Once the resource assignments have been made estimates can be developed for each task associated with the iteration. The detailed estimates are determined by refining the initial capacity-based estimates through the use of traditional estimating techniques that can be augmented through a commitment-driven approach. The commitment-driven estimating approach involves asking those team members assigned to provide an estimate of how much effort will be required to complete the tasks for each requirement in the iteration.

The individual task estimates balanced with the team velocity form the basis for the estimates for the iteration plan. These estimates should be verified against broader estimates completed as part of the initial planning effort for the iteration.

Identify Dependencies

OUM strives to minimize dependencies within the detailed iteration plans due to the limited duration of the plans. The key to reducing dependencies is to select relatively independent bits of functionality that can then be integrated to achieve the iteration's objectives. However, dependencies cannot be completely eliminated and must be taken into account. This is where the project manager must work with the project team and stakeholders to do further analysis of the MoSCoW List. The dependencies for the use cases addressed in the current iteration must be reflected on the MoSCoW List must also be reflected in the detailed plan for the iteration.

Establish Success Criteria for the Iteration

Each objective must be associated with an evaluation criterion that defines how the achievement will be measured and the levels of those measurements that will be considered a success. These evaluation criteria must be clear and not subjective. The evaluation criteria must be objective and measurable so that progress toward the project's goals can be demonstrated at the end of the iteration.

Determine the Assessment Milestones for Each Iteration

Iteration assessments are essential since iterative development is a collaborative endeavor between the project team, end users and project sponsors. The iteration assessments are important events since they uncover potential issues that could derail the iteration. They also allow the team the ability to take necessary corrective action to get the project righted. Iteration assessments include demonstrating evidence of results, open and honest review of the iteration, discussion on how to improve the next one and acceptance reviews to ensure objectives have been met and the evaluation criteria have been satisfied.

Compile the Iteration Plan

The iteration plan information should be compiled into a simple document. The first section of the plan should reflect the scope and objectives of the iteration. The primary goal of the scope is to make clear the iteration's objectives, priorities and evaluation criteria. This information can be used to track the iteration's progress and drive the iteration assessment. The rest of the plan captures the detailed plan for the execution of the iteration.

Iteratively Evolve the Plan(s)

OUM project planning requires the project manager to plan the iteration by specifying the goals and then allowing the team to determine how best to achieve the goals. More information will be brought to light during the execution of an iteration that will impact the current iteration plan as well as the plan for the next iteration.

The iteration plan may need to be adjusted to account for a previously unknown requirement and/or risk. However, it is recommended that rather than re-planning the current iteration, it is important to keep the team focused on the goal of the current iteration and remove as many barriers as possible to allow the team to move forward.

Also, keep in mind that each of the layers – implementation and iteration plans must be kept in synch. The degree to which an iteration achieves its objectives will impact future iterations as well as milestones on the implementation plan. As the phase and iteration assessments are conducted, feedback will be gained which will lead to the need to adjust the plan(s).

CONSIDERATIONS FOR LARGER PROJECTS

In a large and/or complex OUM project, there may be several product evolutions – one for each major release of the product to be developed, implemented or upgraded. Projects which are more than a year in duration, those with high risk factors and/or projects where there is business value to be gained from delivery of a sub-set of the overall functionality are all candidates for the evolutionary approach. This approach allows risk to be spread over a number of evolutions and permits business value to be delivered sooner.

In the case where the multiple evolutions will be used to accomplish the project's objectives, an overall plan should be established to map out the evolutions. Each evolution should be planned so that the delivery some subset of the project benefits are realized through the delivery of working software and/or implementation of Commercial Off-The-Shelf Software (COTS). This is done by examining the business drivers and schedule goals for the project, laying out the work and assigning the milestones in the roadmap to one or more evolutions. An example evolution roadmap for a planning application is shown in Figure 5 below.

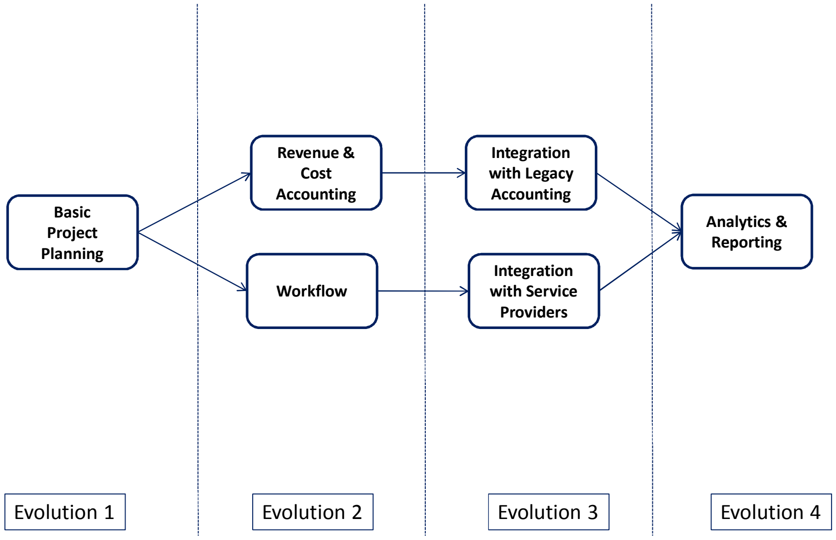


Figure 5: Example Evolution Roadmap

Once the overall plan for the evolutions is established, the planning details are pushed to the lower level implementation and iteration plans for each evolution. As the project progresses through the first evolution and subsequent evolutions more information is brought to light which will require the plans at each layer to be adjusted.

CONCLUSION

OUM recommends a layered approach to project planning rather than a single, highly detailed plan. Planning and managing at each layer (implementation and iteration) frees the project manager and project team from the need to create and "feed" detailed plans that have a tendency to quickly become out of date and unwieldy. The layered planning is scalable and empowers the team to focus on the key goals, measurements, milestones and controls that enables effective project management.

"All endeavor calls for the ability to tramp the last mile, shape the last plan, endure the last hours toil. The fight to the finish spirit is the one... characteristic we must possess if we are to face the future as finishers."

- ***Henry David Thoreau***

.

REFERENCES

This white paper was made possible through the collaboration with numerous OUM contributors as well as the conglomeration of information from several outstanding books and websites:

- Bittner, Kurt. 2007. *Managing Iterative Software Development Projects*. Boston, MA: Pearson Education, Inc.
- Boehm, B. and R. Turner. 2004. *Balancing Agility and Discipline: A Guide for the Perplexed.* Boston, MA: Addison-Wesley
- Cockburn, A. 2002. *Agile Software Development.* Boston, MA: Addison-Wesley.
- Larman, Craig. 2004. *Agile and Iterative Development: A Manager's Guide.* Boston, MA: Pearson Education, Inc.
- Cohn, Mike, 1962 -. *Agile estimating and Planning*, Upper Saddle River, NJ: Pearson Education, Inc.
- A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fourth Edition, Newton Square PA: Project Management Institute, Inc.
- http://www.agilemanifesto.org.
- http://alistair.cockburn.us
- http://www.controlchaos.com/
- http://martinfowler.com/articles/itsNotJustStandingUp.html
- http://Agilesoftwaredevelopment.com/scrum

Planning an OUM Project
February 2011

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com